



UNIDADE CURRICULAR: Sistemas Operativos

CÓDIGO: 21111

DOCENTE: Paulo Shirley

A preencher pelo estudante

NOME: Claudia Andreia Dias Pires

N.º DE ESTUDANTE: 1303334

CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 14ABR2019

RESOLUÇÃO:

O programa começa por validar o número de parâmetros de entrada e se estão de acordo com as regras do enunciado. Se os parâmetros forem inválidos o programa sai em erro e apresenta a mensagem a indicar a razão, caso contrário irá passar o controlo para a função mpss.

A função mpss começa por mostrar o cabeçalho da execução e de seguida inicializa o ficheiro das somas parciais.

A inicialização do ficheiro de soma das execuções consiste em escrever no ficheiro o elemento neutro da soma (o valor 0) como resultado da soma parcial dos np processos que iremos ter. Tal foi necessário pois o processo np poderá escrever no ficheiro o valor da sua soma parcial apos o processo np - 1 dependendo da carga do cpu que esta a executar o cada um dos processos, levando a problemas de concorrência que poderiam originar a perda de algum dos valores parciais, ou mais gravemente, a perda e o registo de um valor parcial irreal.

Apos iniciar o ficheiro lança o ciclo de processamento de somas parciais. Quando o ciclo termina, a função mpss vai calcular a resultado do processamento percorrendo o ficheiro dos resultados parciais e acumulando cada uma das entradas, de seguida vai fazer o mesmo cálculo utilizando a expressão:

$$s(n) = [n (n + 1) (2n + 1)] / 6,$$

após mostrar o resultado ao utilizador termina com sucesso.

O ciclo de processamento utilizado para gerir/criar os processos dedicados de 1 a np inicia-se com a criação de um processo filho (com a excepção do último processo, já que o mesmo não tem filhos), de seguida o processo calcula a sua soma parcial recorrendo a um ciclo for. De seguida o processo escreve a sua soma parcial no ficheiro auxiliar na posição alocada para o efeito. imprime uma mensagem com o seu nº de ordem, PID, PPID, intervalo de valores de i para o qual calculou a soma parcial e o valor dessa soma e espera que o seu filho termine (excepto o se for o último processo, pois não tem filho) e ele próprio termina.

DECISÕES DE IMPLEMENTAÇÃO:

init_file: De modo a otimizar a escrita no ficheiro foi utilizado um vector de 32 zeros, já que era o valor máximo de np, mas apenas é escrito o número de zeros iguais ao valor de np que foi passado por parâmetro, desta forma garantiu-se que apenas existiam np zeros no ficheiro.

calcula: A função calcula abre o ficheiro dos resultados parciais em modo leitura binária e enquanto existirem valores no ficheiro, lê e soma-os, devolvendo o resultado da soma.

mpss_parcial: A função começa por calcular o índice do ficheiro onde irá escrever o resultado do processamento, bem como o início e o fim do bloco a processar. Caso não seja o último bloco do processamento irá lançar um processo de modo a processar o bloco seguinte e após processar o seu bloco ficará a espera que ele termine. A função apresenta um comportamento recursivo, onde se lança a ela própria de modo a processar blocos, e quando o processamento do último bloco ocorre, a função começa a retornar em cascata. O resultado do processamento do bloco é guardado no ficheiro "data.aux", no índice respeitante ao bloco a processar.

mpss: A função principal deste programa, que inicializa o ficheiro de resultados, imprime o cabeçalho e lança o primeiro bloco (invocação á função mpss_parcial) que irá despoletar o processamento em cascata. Após o processamento terminar vai calcular e apresentar os resultados do processamento.

EXEMPLO:

./mpss 14 100

```
mpss
Soma do quadrado dos primeiros 660 numeros naturais com cadeia de 14 processos d
edicados
Processo inicial P0: PID= 2861 PPID= 2860
Processo P1 : PID= 2862 PPID= 2861 i=[ 1- 47] soma= 35720
Processo P2 : PID= 2863 PPID= 2862 i=[ 48- 94] soma= 245575
Processo P3 : PID= 2864 PPID= 2863 i=[ 95-141] soma= 663076
Processo P4 : PID= 2865 PPID= 2864 i=[142-188] soma= 1288223
Processo P5 : PID= 2866 PPID= 2865 i=[189-235] soma= 2121016
Processo P6 : PID= 2867 PPID= 2866 i=[236-282] soma= 3161455
Processo P7 : PID= 2868 PPID= 2867 i=[283-329] soma= 4409540
Processo P8 : PID= 2869 PPID= 2868 i=[330-376] soma= 5865271
Processo P9 : PID= 2870 PPID= 2869 i=[377-423] soma= 7528648
Processo P10 : PID= 2871 PPID= 2870 i=[424-470] soma= 9399671
Processo P11 : PID= 2872 PPID= 2871 i=[471-517] soma= 11478340
Processo P12 : PID= 2873 PPID= 2872 i=[518-564] soma= 13764655
Processo P13 : PID= 2874 PPID= 2873 i=[565-611] soma= 16258616
Processo P14 : PID= 2875 PPID= 2874 i=[612-660] soma= 19830104
Resultado calculado: 96049910
Resultado esperado : 96049910

Process returned 0 (0x0) execution time : 0.003 s
Press ENTER to continue.
█
```